IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kondamudi Raaga Laasya
7th February 2024

https://github.com/RaagaLaasya/DataScience_IBM/tree/main

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - EDA with SQL
  - EDA with data visualization
  - Building an interactive map with Folium
  - Building a dashboard with Plotly Dash
  - Predictive analysis (Classification)

- Summary of all results
  - EDA results
  - Interactive analytics
  - Predictive analysis

Introduction

- Project background and context
  - SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- Problems you want to find answers
  - The project task is to predict if the first stage of the SpaceX Falcon 9 rocket will land successfully.

Section 1

Methodology

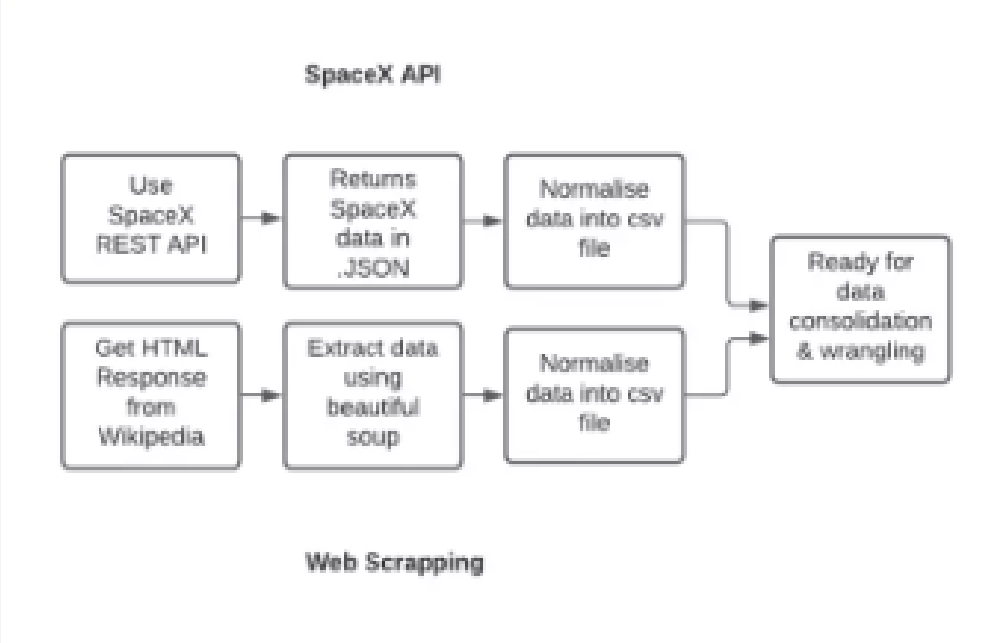
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning and data cleaning of null values and irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, SVM, DT models have been built and evaluated for the best classifier

Data Collection

- Two methods of data collection: using the SpaceX REST API and by Web Scrapping.

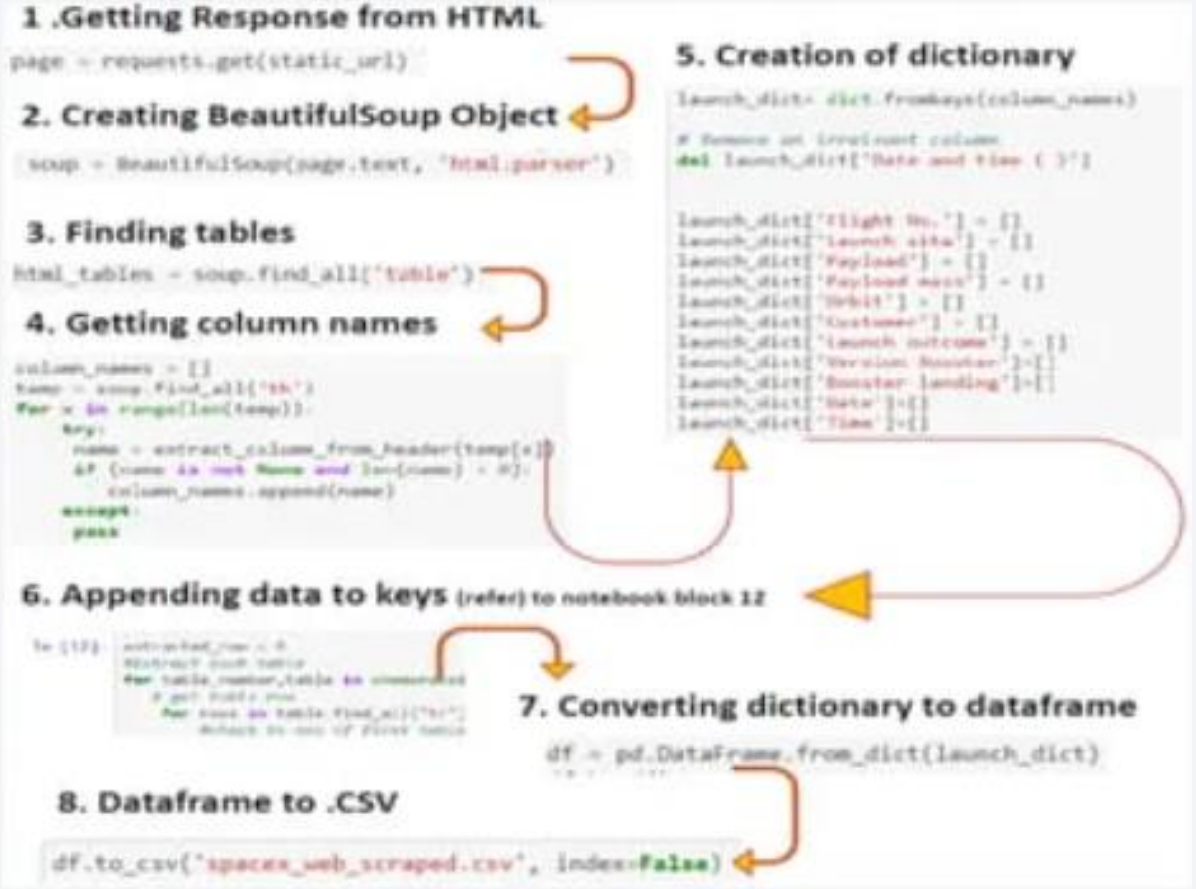


Data Collection – SpaceX API

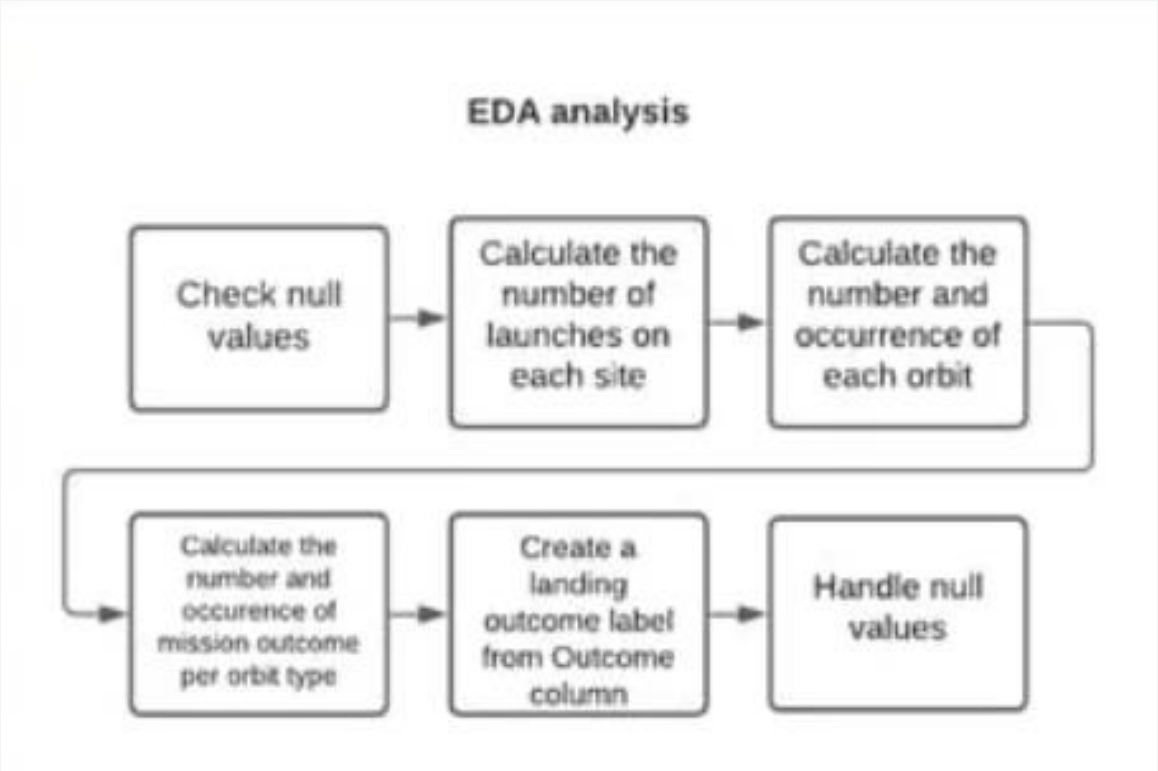
- Data collection with SpaceX REST calls
- https://github.com/RaagaLasya/DataScience_IBM/blob/main/data-collection-api.ipynb

Data Collection - Scraping

- Web scraping process
- https://github.com/RaagaLaasya/DataScience_IBM/blob/main/webscraping.ipynb

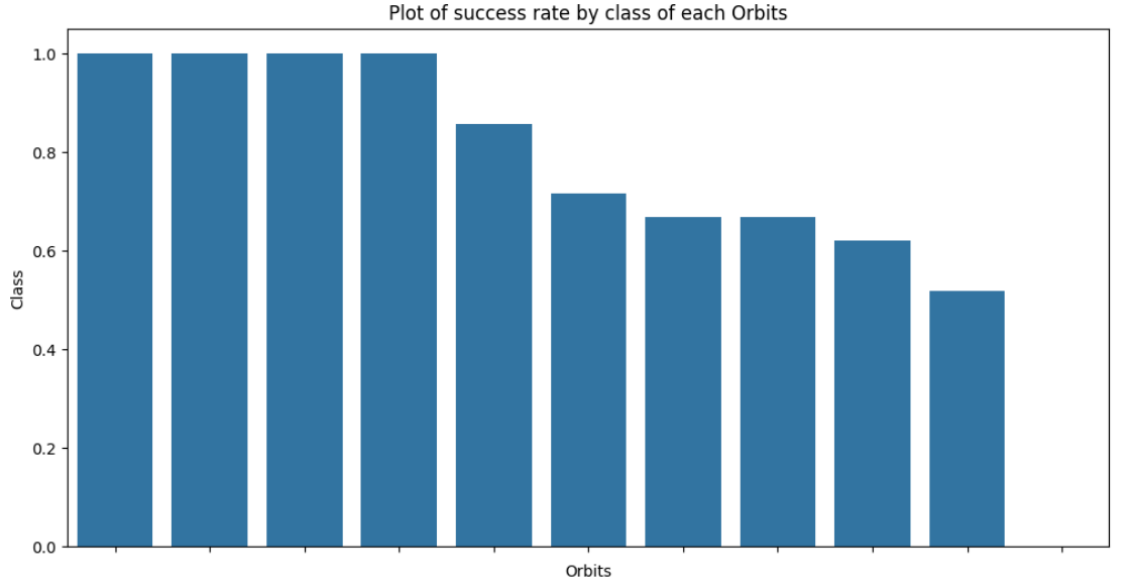


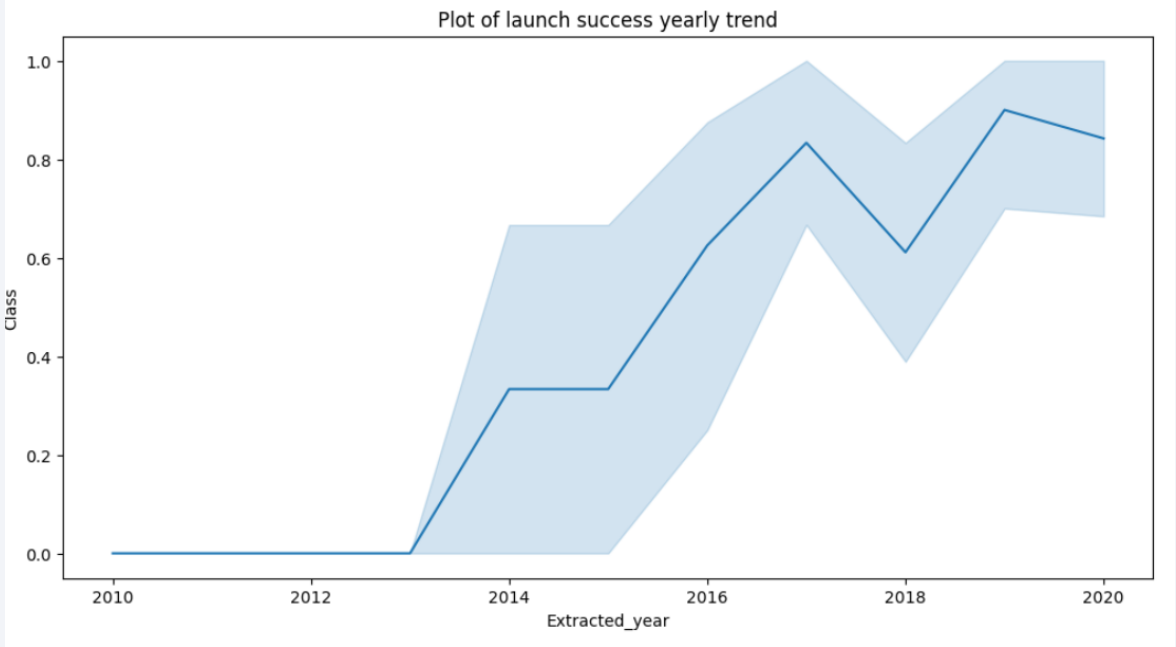
Data Wrangling

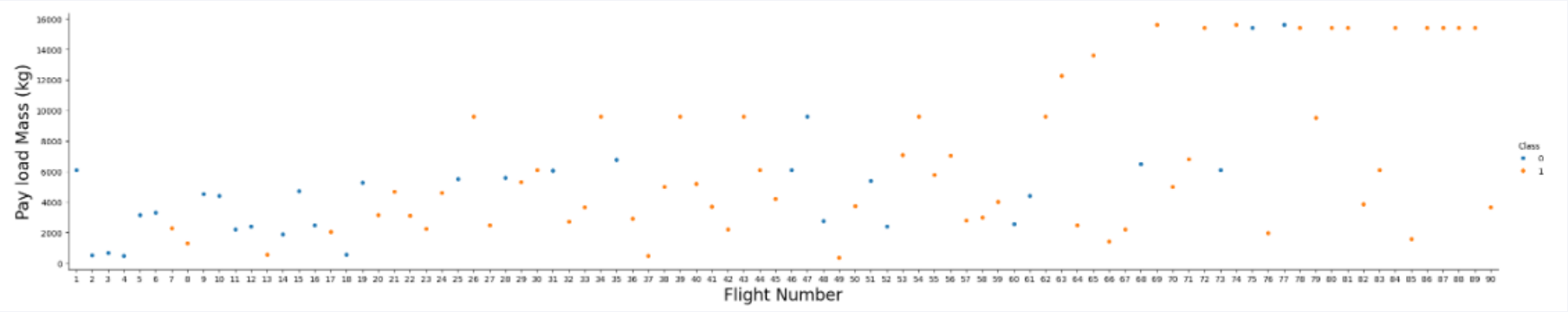


https://github.com/RaagaLaasya/DataScience_IBM/blob/main/Data%20wrangling.ipynb

EDA with Data Visualization







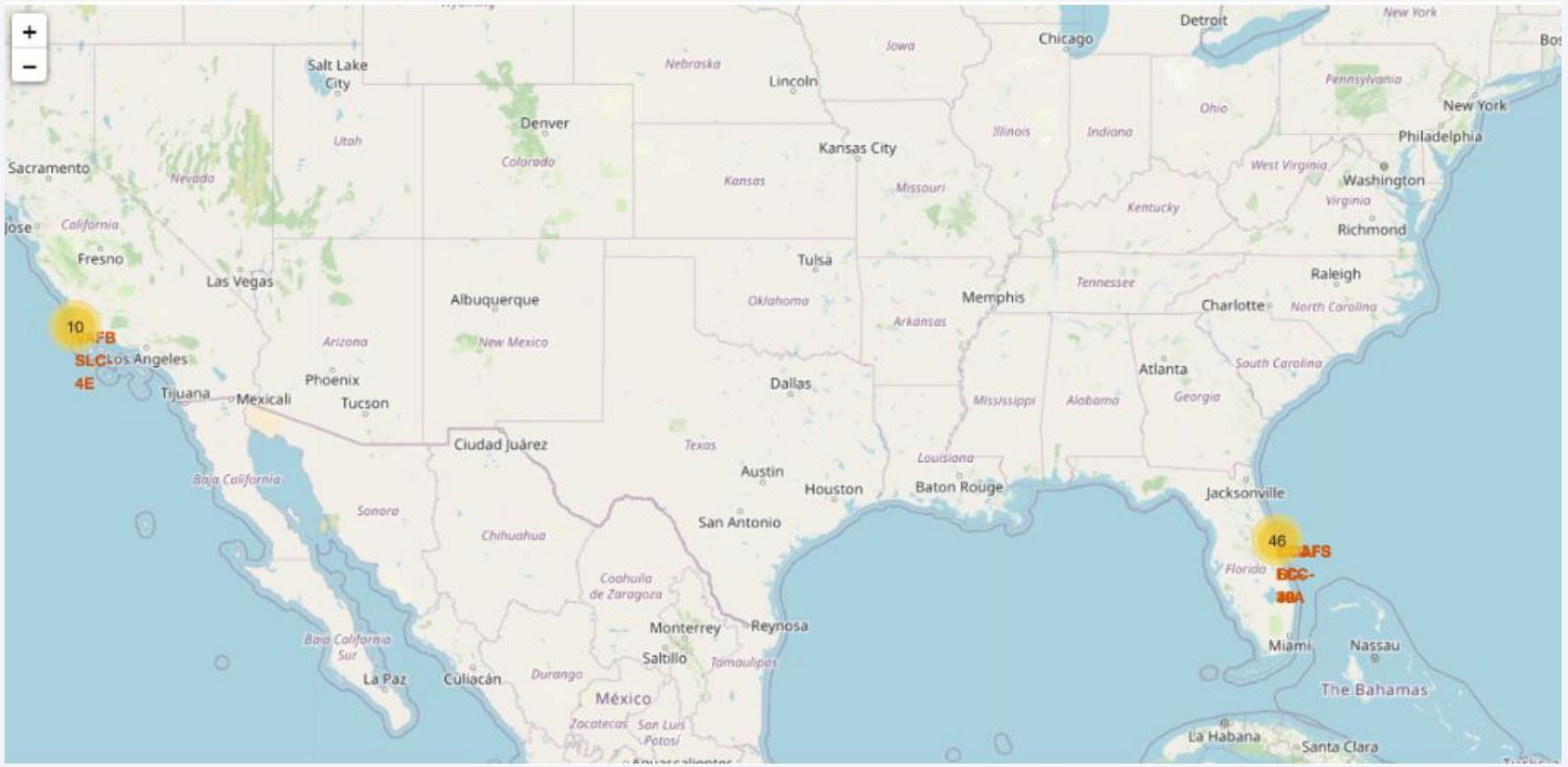
https://github.com/RaagaLaasya/DataScience_IBM/blob/main/eda-dataviz.ipynb

EDA with SQL

https://github.com/RaagaLaasya/DataScience_IBM/blob/main/eda-sql-coursera_sqllite.ipynb

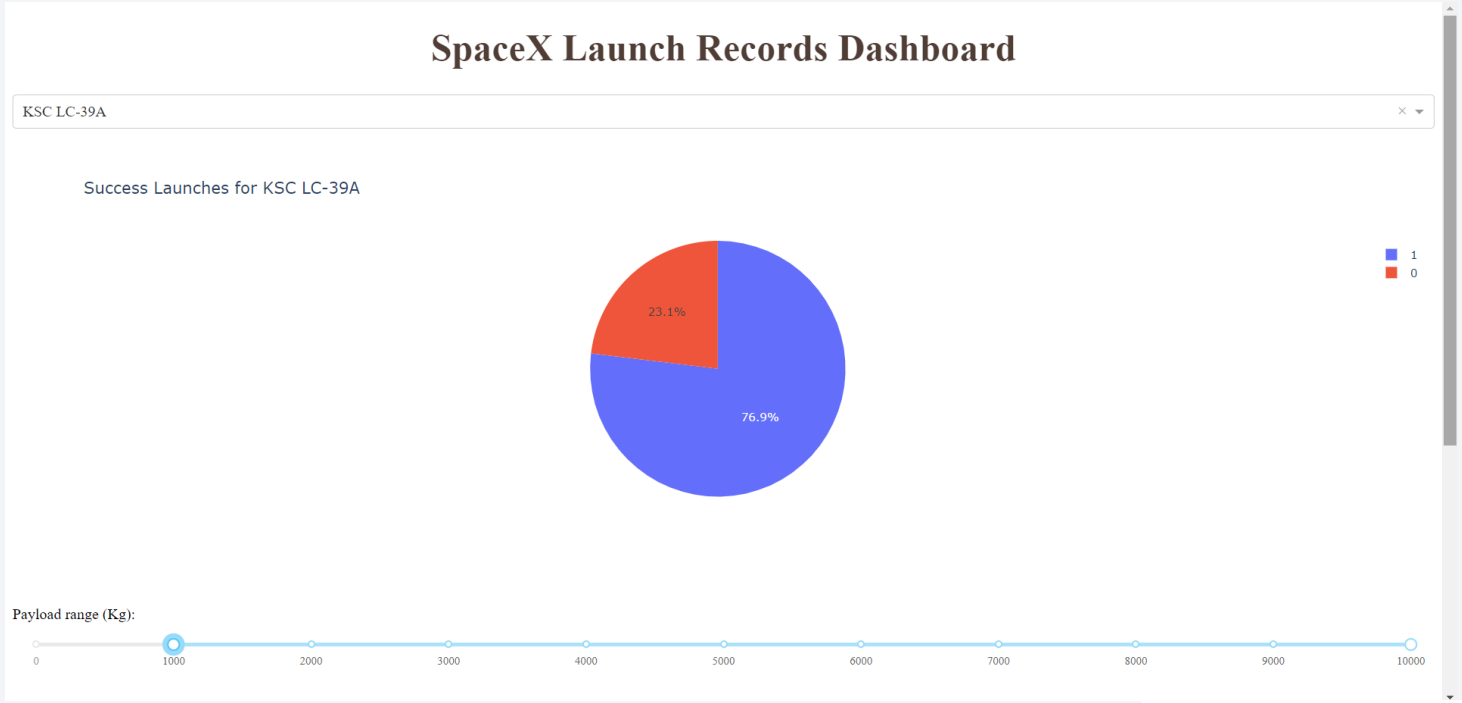
- SQL queries performed include:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first succesful landing outcome in ground pad was acheived.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
  - Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

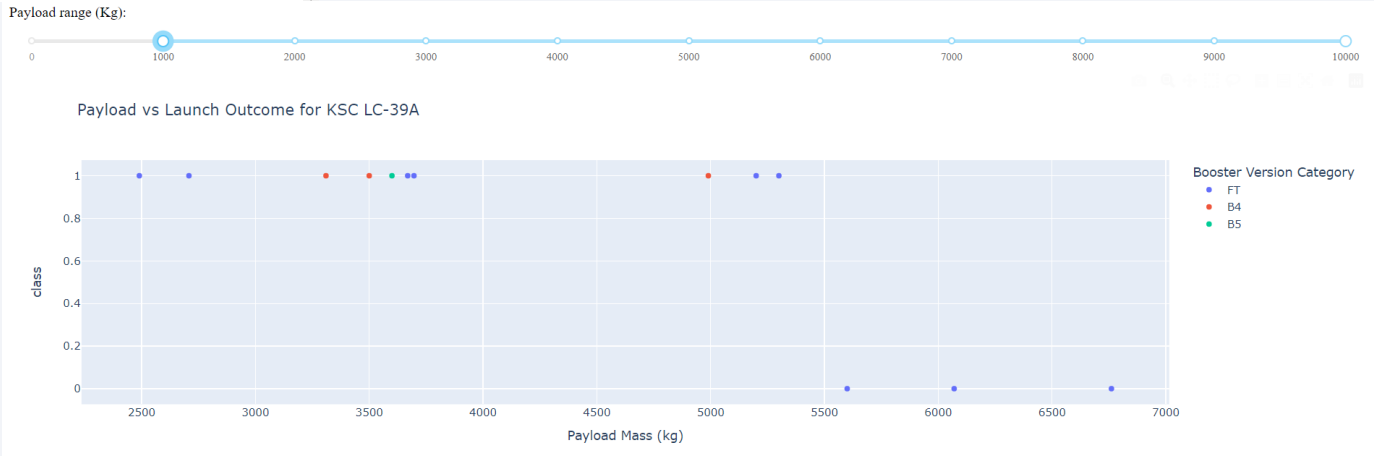
Build an Interactive Map with Folium



Map markers have been added with the aim to find an optimal location for building a launch site https://github.com/RaagaLaasya/DataScience_IBM/blob/main/folium_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

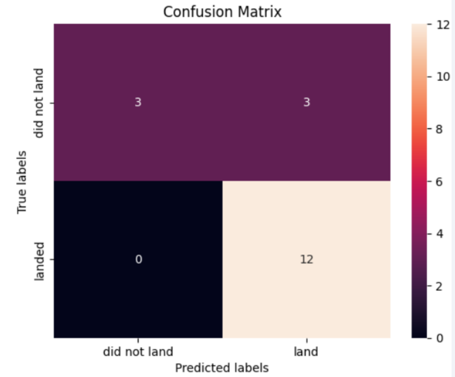


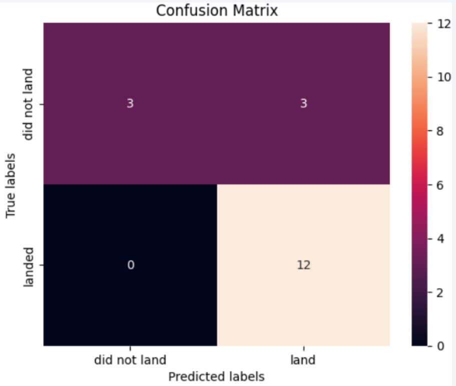


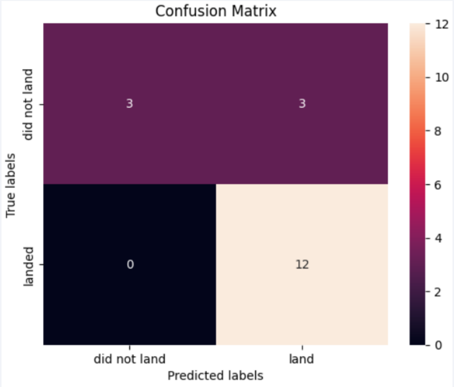
https://github.com/RaagaLaasya/DataScience_IBM/blob/main/spacex_dash_app.py

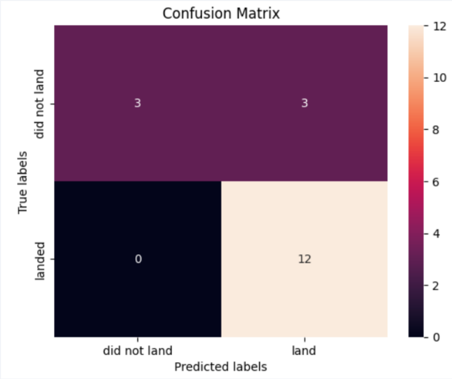
Predictive Analysis (Classification)

- SVM,KNN, Logistic Regression and Decision Tree achieved an accuracy of 83.3% however the best model is Decision Tree with a score of 0.873.







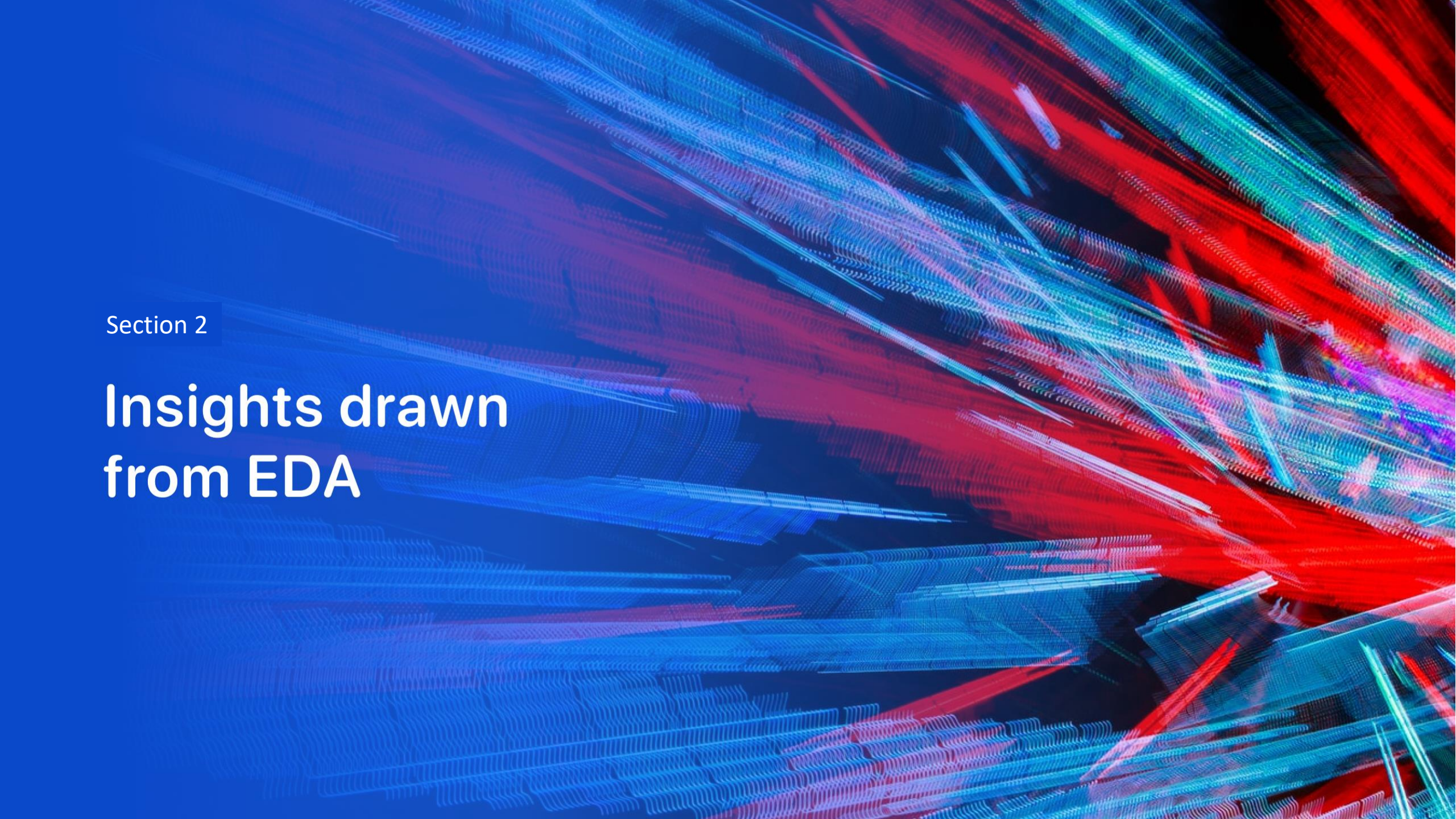



```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5,
'splitter': 'random'}
```

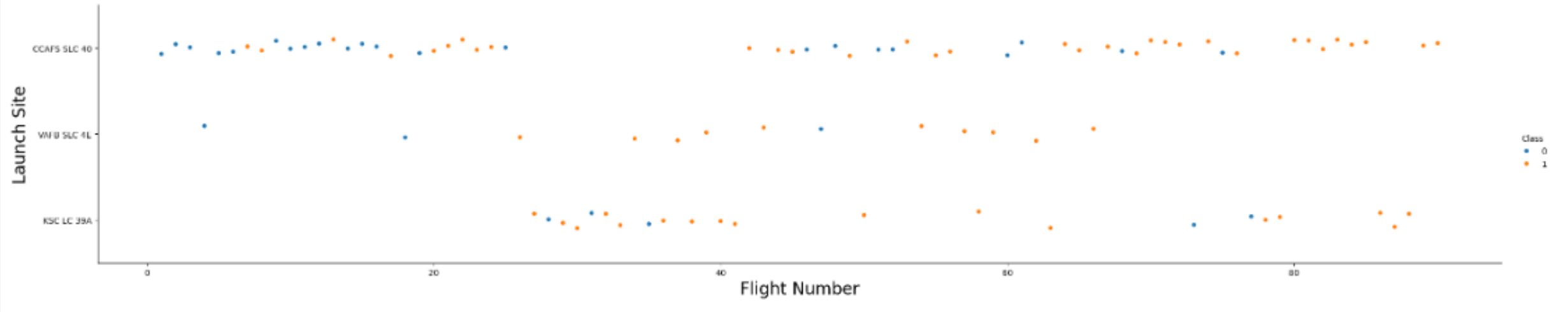
https://github.com/RaagaLaasya/DataScience_IBM/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

- Orbit GEO, HEO, SSO, ES L1 has the best success rate.
- KSC LC 39A had the most successful launches from all sites.
- Low weighted payloads perform better than heavier payloads
- All machine learning models performed very well on the given dataset.

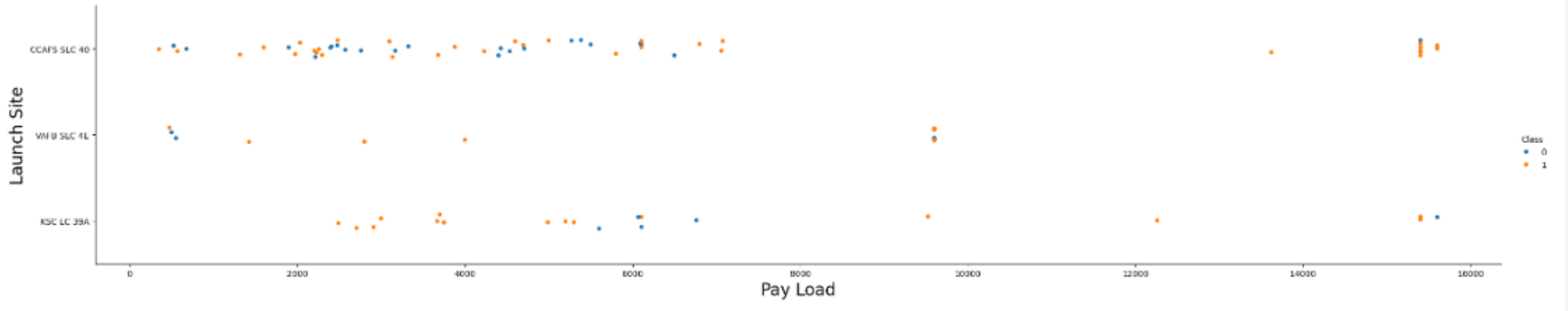

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



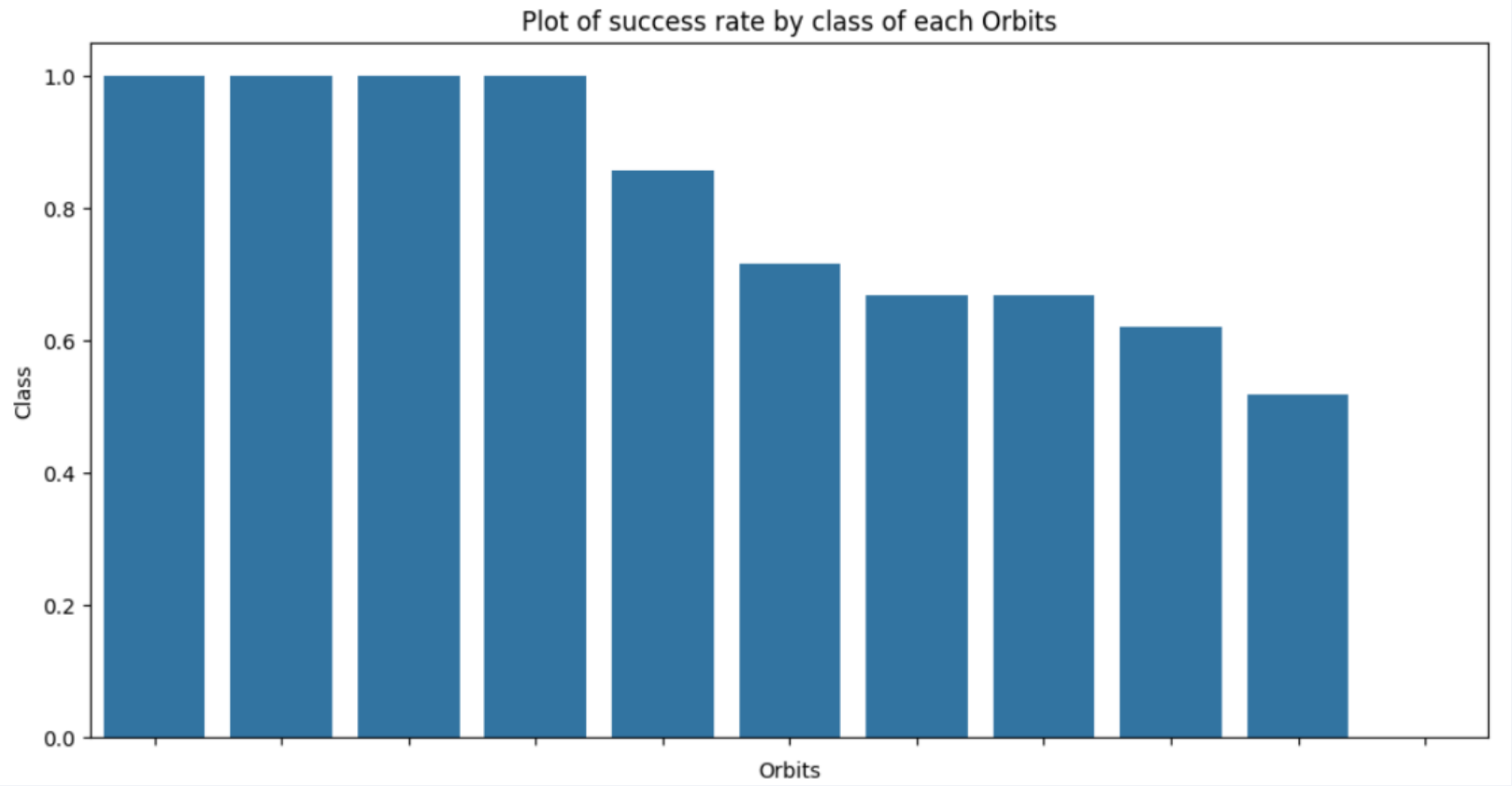
- Launches from the CCAFS SLC40 are significantly higher that the other launch sites.

Payload vs. Launch Site



- The majority of lower mass payloads have been launched from the CCAFS SLC40

Success Rate vs. Orbit Type

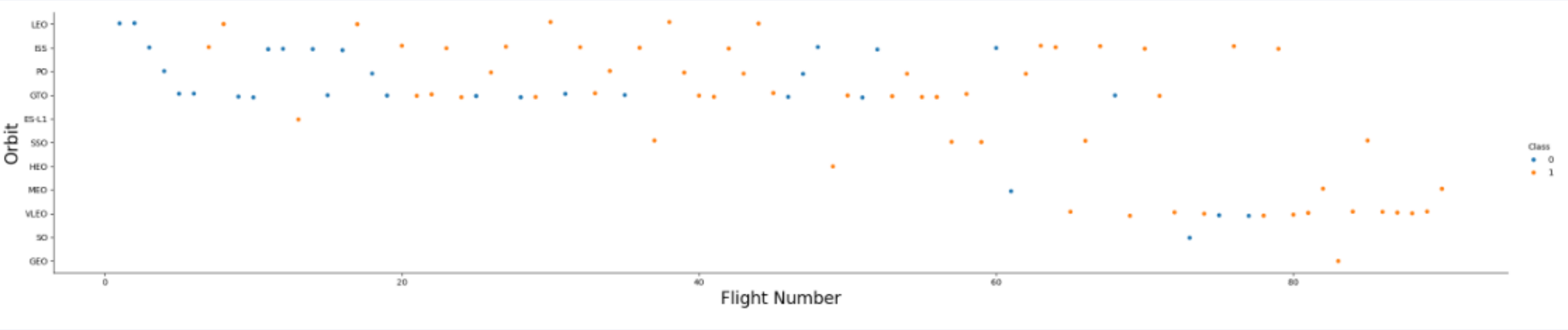



```
df_groupby_orbits = df.groupby('Orbit').Class.mean()
df_groupby_orbits
```

```
Orbit
ES-L1    1.000000
GEO      1.000000
GTO      0.518519
HEO      1.000000
ISS      0.619048
LEO      0.714286
MEO      0.666667
PO       0.666667
SO       0.000000
SSO      1.000000
VLEO     0.857143
Name: Class, dtype: float64
```

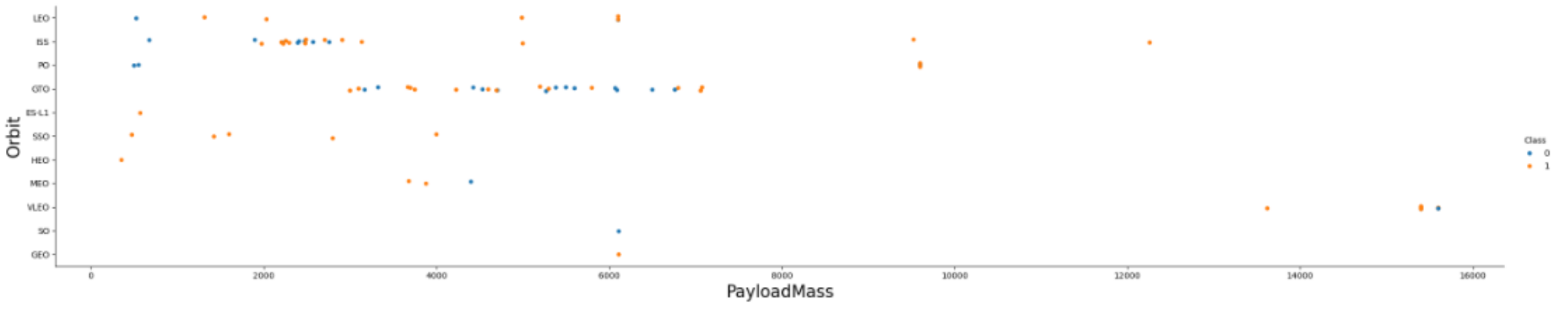
- Orbit types ESL1, GEO, HEO AND SSO have the highest success rate.

Flight Number vs. Orbit Type



- Shift to VLEO is evident

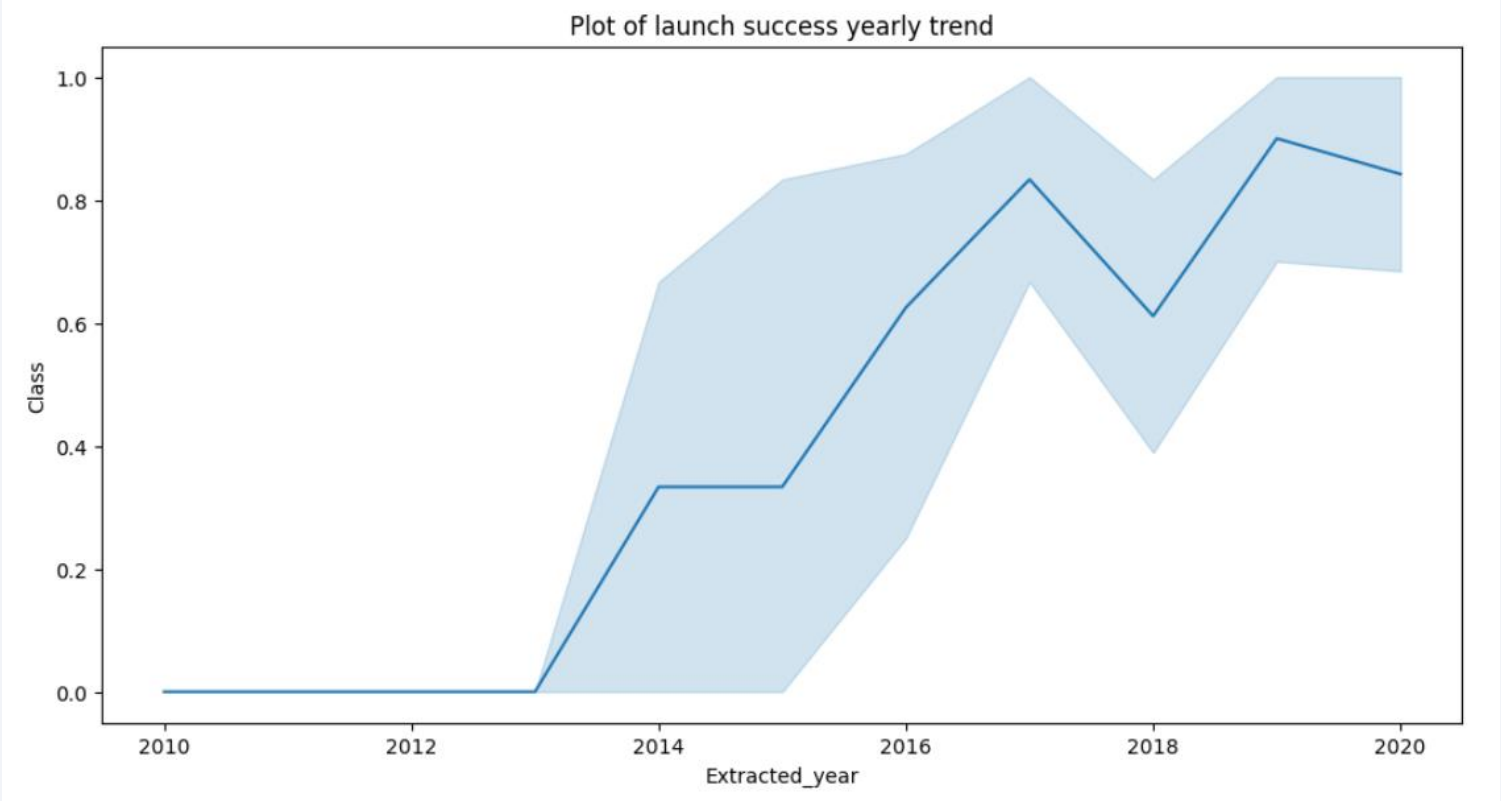
Payload vs. Orbit Type



- Strong correlation between ISS and Payload at the 2000 range, same as GTO and 4000-8000 range

Launch Success Yearly Trend

- Launch success has increased since 2013 and stabilized in 2019, due to advance in science and technology.



All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

In [13]:
```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
```

```
 * sqlite:///my_data1.db
Done.
```

Out[13]:

Launch_Site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string 'CCA'

In [14]:
```
sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * sqlite:///my_data1.db
Done.
```

Out[14]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

In [15]:
```
sql SELECT SUM (PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

Out[15]:

SUM (PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

Task 4

Display average payload mass carried by booster version F9 v1.1

In [16]:
```
sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1%'
```

```
 * sqlite:///my_data1.db
Done.
```

Out[16]:

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

In [28]:
```
sql SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

```
 * sqlite:///my_data1.db
Done.
```

Out[28]:

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [29]:
```
sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING_OUTCOME = 'Succ
```

```
 * sqlite:///my_data1.db
Done.
```

Out[29]:

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [23]:
```
sql SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEXTBL GROUP BY MISSION_OUTCOME
```

```
 * sqlite:///my_data1.db
Done.
```

Out[23]:

Mission_Outcome	COUNT(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [26]:
```
sql SELECT BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SP
```

```
 * sqlite:///my_data1.db
Done.
```

Out[26]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [30]:
```
sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING_OUTCOME='Failure (drone ship)' AND DATE LIKE '2015%';
```

```
 * sqlite:///my_data1.db
Done.
```

Out[30]:

Booster_Version	Launch_Site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

In [32]:
```
sql SELECT LANDING_OUTCOME, COUNT(*) AS qty FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDINC
```

```
 * sqlite:///my_data1.db
Done.
```

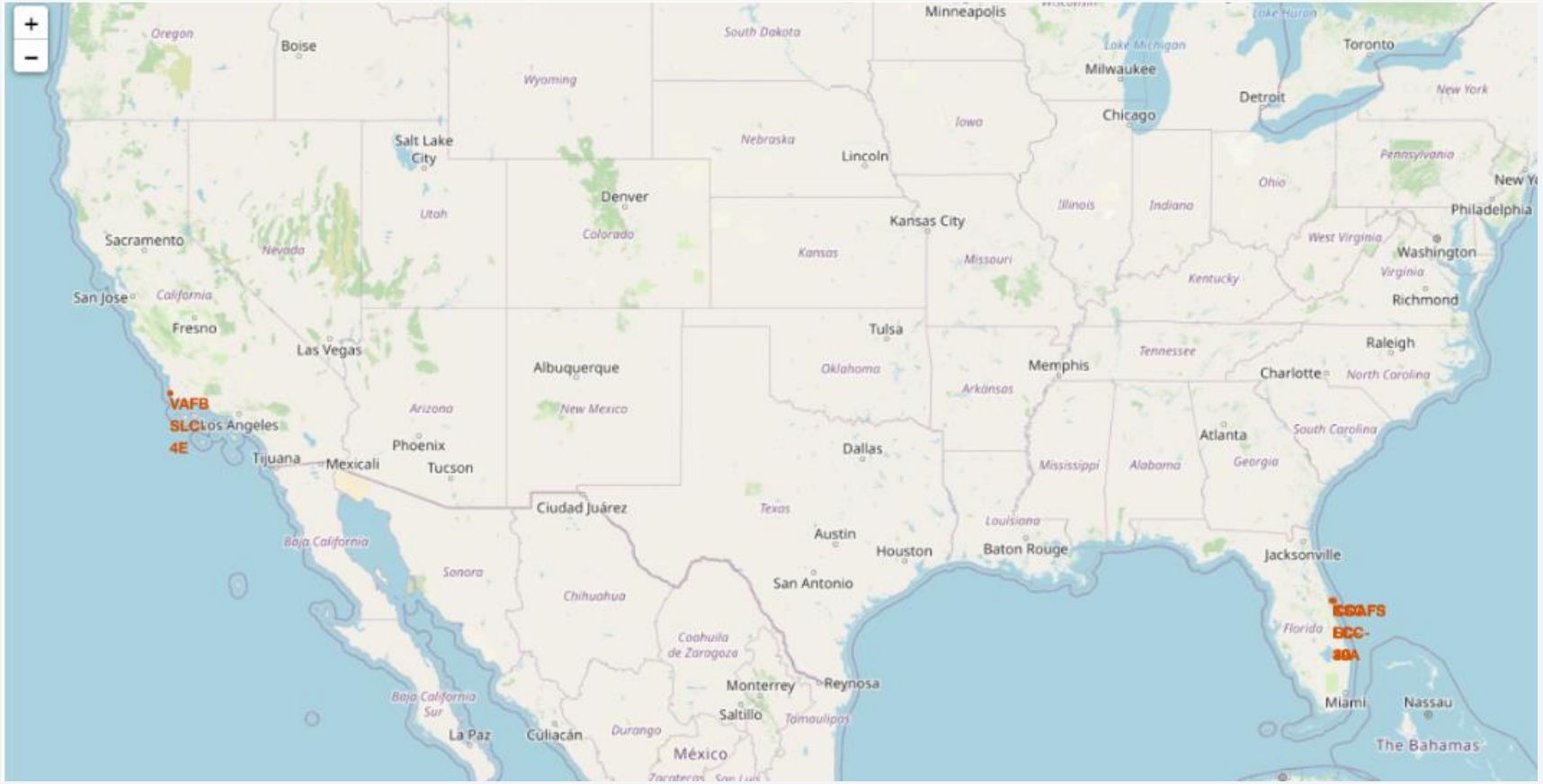
Out[32]:

Landing_Outcome	qty
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

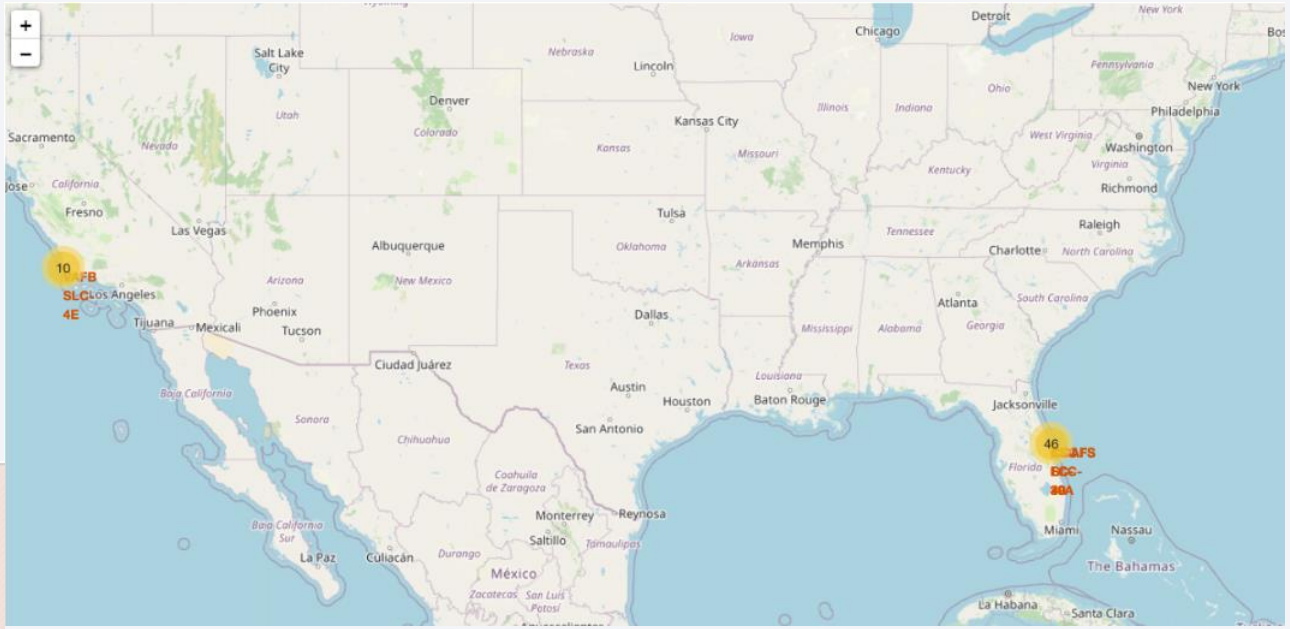
Section 3

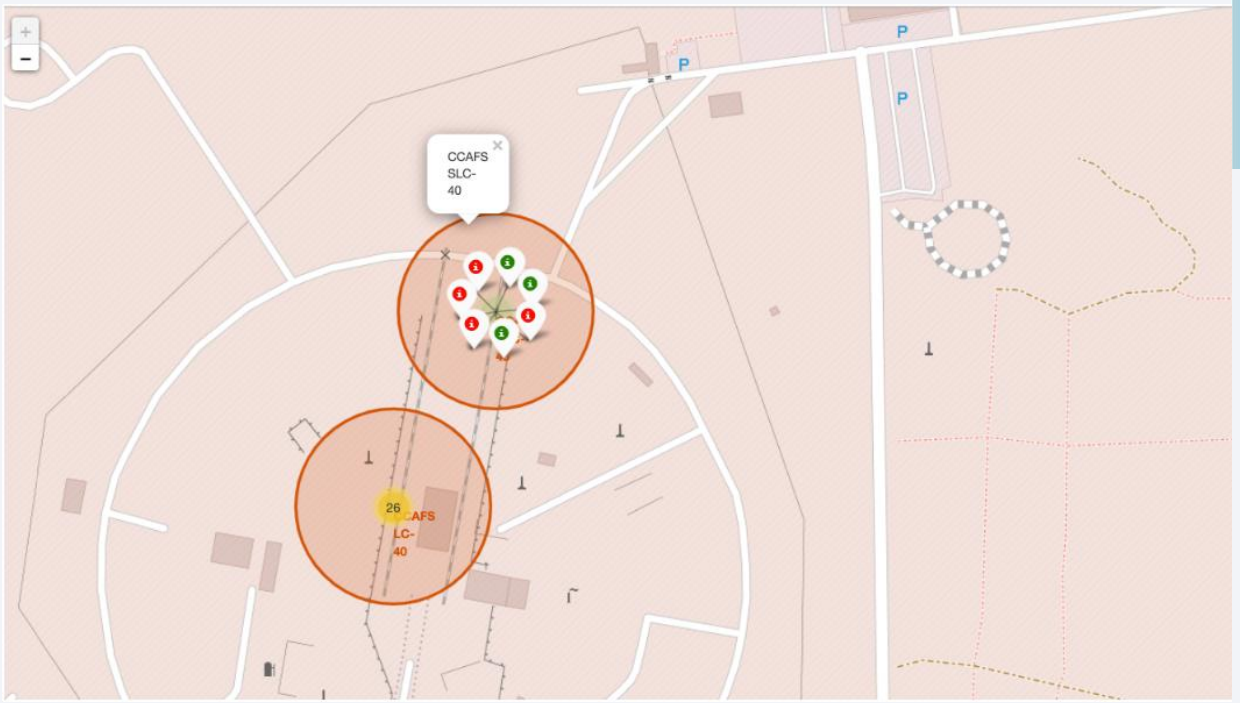
Launch Sites Proximities Analysis

Mark all launch sites in a map

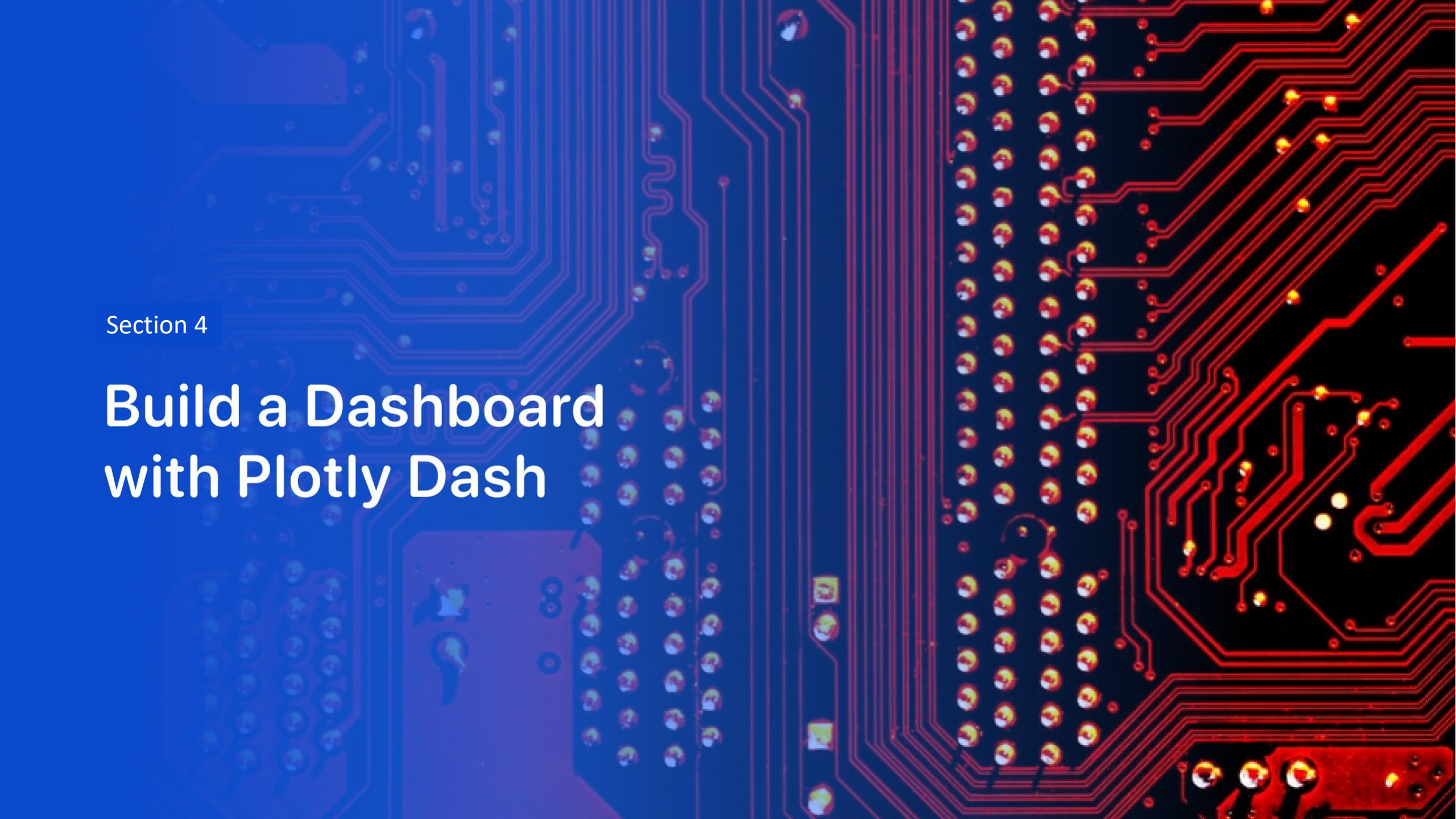


Mark the successful/failed launches for each site on the map

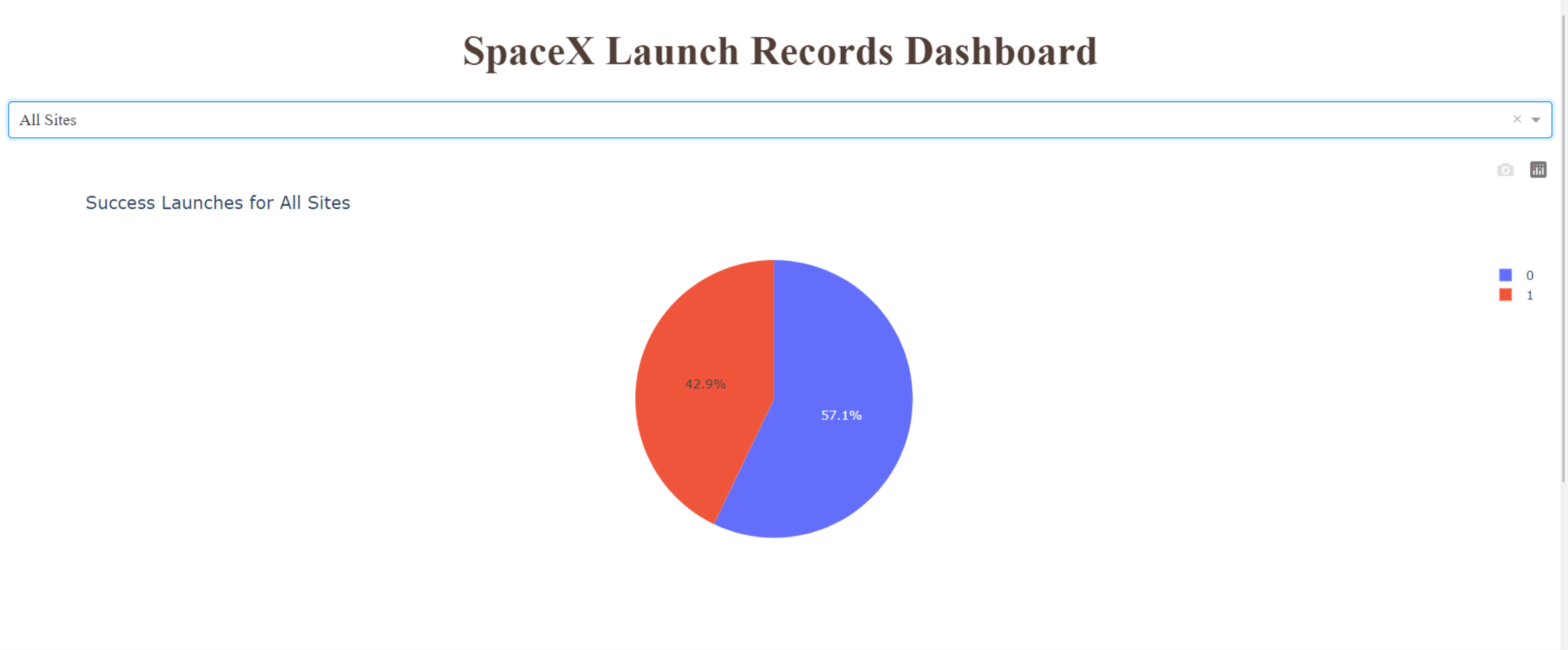




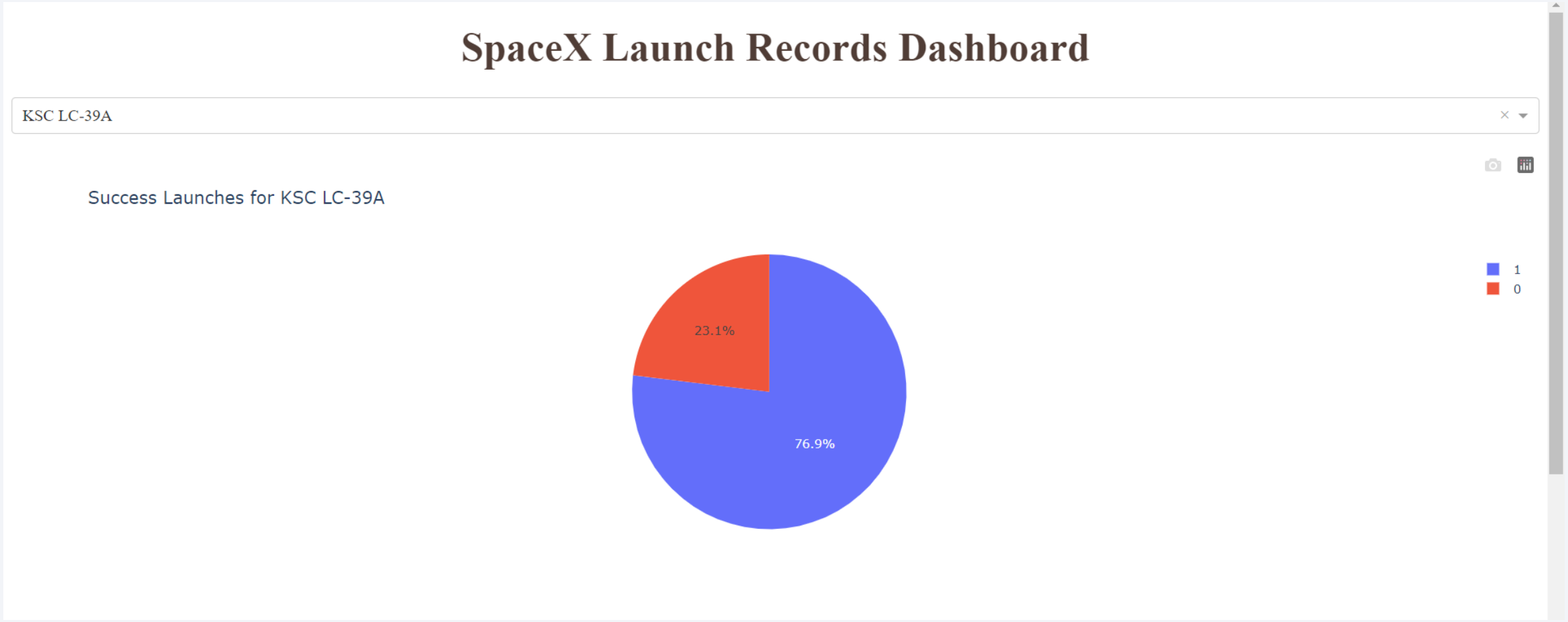
Calculate the distances between a launch site to its proximities


Section 4
Build a Dashboard with Plotly Dash

Total count


SpaceX Launch Records Dashboard
All Sites
Success Launches for All Sites
0
1
42.9%
57.1%

Success rate by site



Payload vs launch outcome

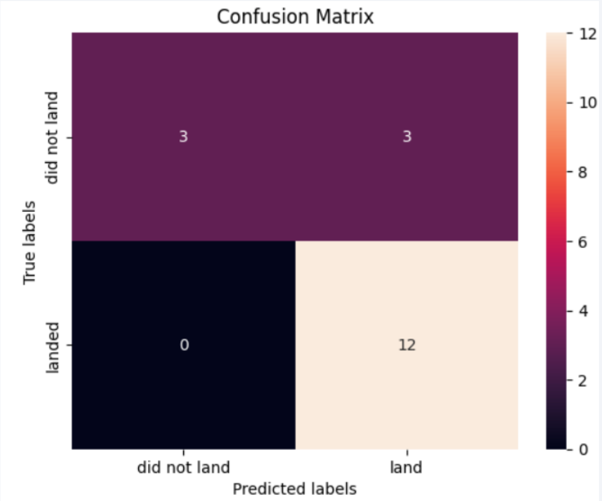
Section 5

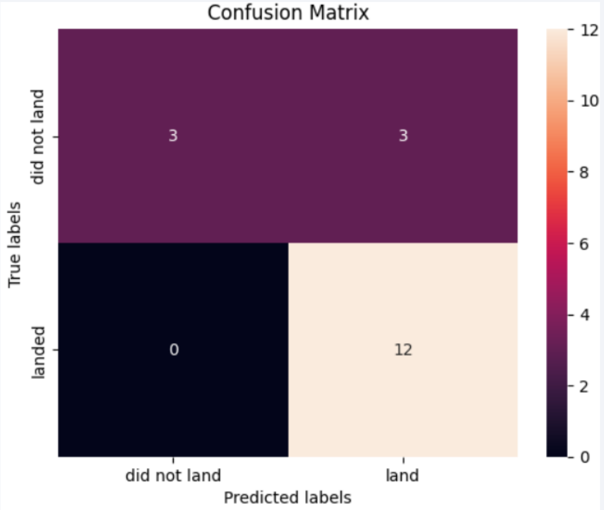
Predictive Analysis (Classification)

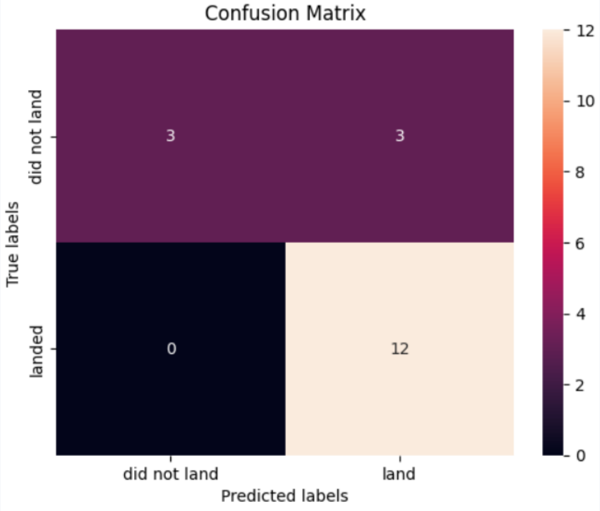
Classification Accuracy

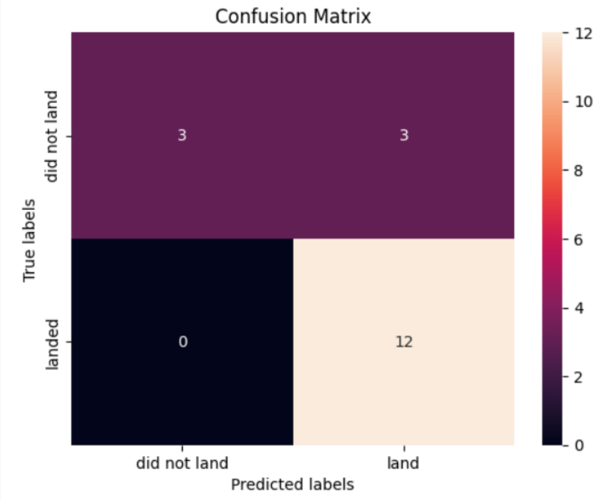
The accuracies of logistic regression, svm, knn and decision tree were all found to be 83.3%, therefore the use of a bar chart would be redundant.

Confusion Matrix









Conclusions

- SVM,KNN, Logistic Regression and Decision Tree achieved an accuracy of 83.3% however the best model is Decision Tree with a score of 0.873

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5,
'splitter': 'random'}
```

Thank you!